Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

A reference model for teleconferencing systems

C. Bonini, W.J. Fokkink and A. Lesch

Computer Science/Department of Software Technology

CS-R9502 1995

Report CS-R9502
ISSN 0169-118X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

A Reference Model for Teleconferencing Systems*

Cinzia Bonini
Intec$_S$ Sistemi
Pisa
Italy
cinzia@pisa.intecs.it

Wan Fokkink
CWI
Amsterdam
The Netherlands
wan@cwi.nl

Arek Lesch
Alcatel/SEL
Stuttgart
Germany
alesch@rcs.sel.de

Abstract

We present a specification in Z of the framework of a teleconferencing system, which combines text, computer graphics, video, audio and other features in a computer display. Teleconferencing systems offer the possibility to organize international meetings via internet.

AMS Subject Classification (1991): 68Q60.
CR Subject Classification (1991): F.3.1.
Key words & Phrases: Teleconferencing system, hypermedia, Z.

1 Introduction

These last few years, there has been an outburst in research concerning *multimedia systems*, which offer the possibility to combine text, computer graphics, video, audio and other features in a computer display (for an overview of the current state of the art, see [Tem94]). Lately, this research has been extended to the field of *teleconferencing systems*, which allow the display of these features on several computer screens at a time, together with an audio connection and possibly a video connection between these computers. Such systems offer a magnificent alternative for the money and time consuming circuit of international conferences. Teleconferencing structures are intrinsically complicated, being a mix of human interaction, multimedia systems, databases and internet.

As usual, the rapid development of teleconferencing systems has lead to a Babylonian use of terminology. Moreover, although the systems that we know of are alike, they all differ in subtle ways, such as how a teleconferencing session is started and ended, how the participants decide on the course of a session, and in which way the results of a session are produced. Therefore, it seems time to try and find a consensus of common sense what are to be the features in a teleconferencing system, how they are to be implemented, and how they interact.

We present a Z specification for the framework of teleconferencing systems. Z is a formal specification language based on set theory and predicate logic. The idea of

*This work has been carried out in the context of RACE project no. 1046, BOOST.

the specification is to give insight in the possible design decisions that have to be taken while constructing a teleconferencing system. Design decisions in this paper are based on practical experiences with several teleconferencing systems, most notably 'Hypermedia System' from Alcatel/SEL [LS90, LRS93, JLM+93] and 'Meeting' from the CWI.

Our work is in the line of the paper on the *Dexter model* for hypertext systems of Halasz and Schwartz [HS90]. They have written a specification for a framework of hypertext systems in Z. Also, they have suggested a standard terminology that has been widely accepted in the field. Our Z specification supplies a basic means for anyone who wants build a teleconferencing system. Furthermore, the specification can serve as a first platform to decide upon a standard terminology for this field of research.

Of course, familiarity with Z will be a great help in trying to read the specification. However, all its schemas are provided with informal comments, which should be sufficient to get a feel for the specification, even without any knowledge of Z. For a clear and concise introduction into the secrets of Z, see [Spi89]. A thorough treatment on the semantics of Z is provided in [Spi88].

We have structured the specification using the ZNICE tool from Intecs, which is a support tool for Z. It provides a language-based editor [RT89] and a type-checker. The editor is designed to make the construction of Z specifications quick and convenient through the use of templates and text editing interface, WYSIWYG (what you see is what you get) and automatic syntax and semantics checking. ZNICE is integrated with HoodNICE, which is a support tool for the HOOD method [Gro92] of object oriented decomposition of a design, based on the identification of objects and operations. Z has been merged into the HOOD method by formulating a software problem as a HOOD system, see [dGI90].

This work originates from the need of Alcatel/SEL to find a rigorous basis for the construction of its Hypermedia System. The knowledge of teleconferencing systems at Alcatel/SEL, the knowledge of Z and formal methods at Intecs and the CWI, and the tool support for Z at Intecs, turned out to be a good synthesis for producing the specification of teleconferencing systems in Z that is presented in this paper. These three components got together in the RACE project BOOST.

Acknowledgements. We thank Lynda Hardman and Jack Jansen from the CWI, Silvia Mazzini and Gaetano Prestia from Intecs, and Peter Szabo from Alcatel/SEL for their helpful comments. We are grateful to the BOOST project, which provided a perfect environment to produce this work.

2 Description of a Teleconferencing Session

We give a short description of a teleconferencing session. At some points, this informal description is more specific then the Z specification, which will be presented in the next section.

A session involves m sites with their own databases DB_i, and with $\mathrm{n}(i)$ users $\mathrm{U}^{\mathrm{i}}_{\mathrm{j}}$, see Figure 1. Each database is divided into a private and a public part. If several sites are involved in the session, then only information from the public domains of the databases

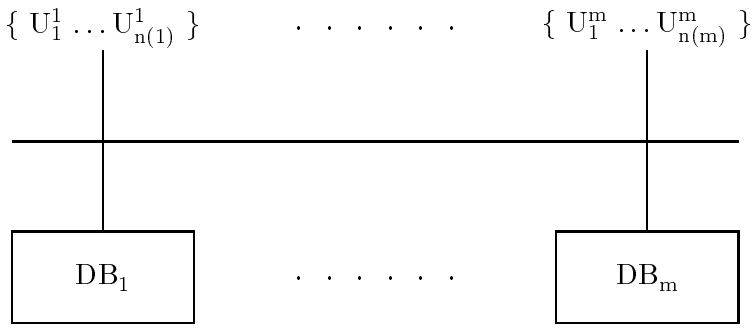

Figure 1: Graphical description of a teleconferencing structure

can be used during the session. If a session is started, each database creates a special directory, which is to contain the products of the session.

Communication between the participants of the session goes via an audio channel, and possibly also via a video channel. Moreover, there is an arrow which can be used (by one participant at a time) to point at objects inside the screen.

The participants may agree on changing the 'product' of the session, which means that one of them, say U, is allowed to change it. U has to wait till all partners have connected to his database, so that they can immediately see all the changes made in the screen by U. When U is finished, the partners have to agree on the changes. If they do so, the changes are stored.

During a session, a new partner may be invited to join, provided that the participants that are already in the session agree on this extension. Since all information produced during the session is contained in one directory, this information can easily be provided to the new participant. At any time during a session, a partner may decide to quit. A session is finished if all participants have quit.

Each participant in the session can decide to start changing the product of the session. Such changes are stored in a temporary directory. At the end of the change, the participants together are to decide whether or not the changes are maintained.

Experience has taught that 'decision algorithms' are a bottle neck in trying to build user-friendly teleconferencing systems. That is, the participants involved in a teleconferencing session are to agree on the actions that are undertaken during the session. Each teleconferencing system seems to want its own method for the participants to decide on these actions. Some want democratic systems, where decisions are made by majorities, some want dictator systems, where one participant decides. In our specification, we propose a mix of both principles; on the one hand, decisions are taken by majorities, on the other hand, it is possible to allow participants the right of veto. That is, such participants alone can avoid that for example the product of the session is changed, or that a new participant is allowed to join in.

During a teleconferencing session, participants should have the opportunity to go their own ways for a while. Firstly, this provides an opportunity for a small group of participants to separate from the others for a while, in order to study a certain problem, or to create a specific product. Secondly, if participants disagree, they can then decide

to create a product for themselves. Therefore, we propose the feature of a commission, which consists of a select group of participants. They can produce results independently from the main session, and they are provided with their own arrow. In a sense, each commission is a teleconferencing session on its own. At any time, the participants of such a commission can decide to merge into the main session again.

3 The Z Specification

This section presents the Z specification of the framework of a teleconferencing system.[1] We assume that the tool-kit of standard mathematical definitions in Z has been implemented already. See [Spi89, Chapter 4] for a description of this tool-kit.

We start from given finite collections of available participant names $PPANT$ and of data-files $DFILE$. Furthermore, we assume the set $YESNO$ thus defined $\{yes, no\}$.

$$[PPANT, DFILE]$$

$$YESNO ::= yes \mid no$$

3.1 Participants

The first schema *Participant* specifies the participants in a teleconferencing session together with their databases. In a session, each participant $ppant\,(i)$ is connected to a database $dbase\,(i)$. This expression describes only the public domain of the database of $ppant\,(i)$, which is accessible to the other participants. The notation $\rightarrowtail\!\!\!\!\!\!\!+\ $ means that the map $ppant$ is a partial injection, and the notation $\nrightarrow$ means that the map $dbase$ is partial, since there are only finitely many participants and databases involved in a teleconferencing session.

The natural number nop represents the *n*umber *o*f *p*articipants in a particular session. Finally, $known$ denotes the collection of participants that are present in a session.

Participant

$$ppant : \mathbb{N}_1 \rightarrowtail\!\!\!\!\!\!\!+\ PPANT$$
$$dbase : \mathbb{N}_1 \nrightarrow \mathbb{P}\ DFILE$$
$$known : \mathbb{P}\ PPANT$$
$$nop : \mathbb{N}$$

$$\forall\, i, j : \mathrm{dom}\, ppant \mid i \neq j \bullet ppant(i) \neq ppant(j)$$

$$known = \mathrm{ran}\, ppant$$

$$nop = \#(\mathrm{dom}\, ppant)$$

$$\mathrm{dom}\, ppant = 1 \mathrel{..} nop$$

$$\mathrm{dom}\, dbase = 1 \mathrel{..} nop$$

[1]The lay-out of the schemas has been produces by means of a special package of LaTeX macros for printing Z [Spi87].

In the schema *AdaptDatabase* it is described that the database of a participant can be updated during a teleconferencing session. For instance, a participant can move information from the private domain to the public domain of his database, so that it can be applied in the session.

The input *ppant*? in the schema represents the label of the participant who adapts its database, and the input *dbase*? denotes the result of this adaption. The schema *AdaptDatabase* adapts the state space of *Participant*, which is expressed by $\Delta Participant$.

AdaptDatabase

$$\Delta Participant$$
$$ppant? : \mathbb{N}_1$$
$$dbase? : \mathbb{P}\ DFILE$$

$$ppant? \in \operatorname{dom} ppant$$
$$dbase' = dbase \oplus \{ppant? \mapsto dbase?\}$$
$$ppant' = ppant$$
$$known' = known$$
$$nop' = nop$$

3.2 Decisions

Decisions during a session are taken by the participants together. The function *perc* in the schema *Decision* yields the percentage of participants that have to be in favour of a certain decision in order to take it. The collection *veto* denotes those participants that have a right of veto, i.e. their refusal is sufficient to reject a proposal.

The schema *Decision* incorporates the schema *Participant*.

Decision

$$Participant$$
$$perc : \mathbb{N}$$
$$veto : \mathbb{P}\ PPANT$$

$$perc \leq 100$$
$$veto \subseteq known$$

3.3 Changing vetos

Participants can be provided with a right of veto, which is specified in the schema *AddVeto*. First, a group *ppants*? of participants claims a right of veto. The participants together decide whether or not this claim is awarded. The percentage of participants that agree must be at least *perc*, and participants that already have obtained a right of veto must all agree with the claim.

Even so, in the schema *RemoveVeto*, the participants together can decide to take away the right of veto from the participants *ppants*?. In this procedure, a participant with a right of veto cannot avoid, by means of his veto, that his veto is removed.

AddVeto

$$\Delta Decision$$
$$ppants? : \mathbb{P}\ PPANT$$
$$decide? : \text{seq}\ YESNO$$

$$ppants? \subseteq known$$

$$ppants? \cap veto = \varnothing$$

$$\{i : \mathbb{N}_1 \mid decide?(i) = no \bullet ppant(i)\} \cap veto = \varnothing \land$$
$$100 * \#\{i : \mathbb{N}_1 \mid decide?(i) = yes \bullet ppant(i)\} \geq perc * nop \Rightarrow$$
$$\quad veto' = veto \cup ppants?$$

$$\{i : \mathbb{N}_1 \mid decide?(i) = no \bullet ppant(i)\} \cap veto \neq \varnothing \lor$$
$$100 * \#\{i : \mathbb{N}_1 \mid decide?(i) = yes \bullet ppant(i)\} < perc * nop \Rightarrow$$
$$\quad veto' = veto$$

$$perc' = perc$$

RemoveVeto

$$\Delta Decision$$
$$ppants? : \mathbb{P}\ PPANT$$
$$decide? : \text{seq}\ YESNO$$

$$ppants? \subseteq veto$$

$$\{i : \mathbb{N}_1 \mid decide?(i) = no \bullet ppant(i)\} \cap (veto \setminus ppants?) = \varnothing \land$$
$$100 * \#\{i : \mathbb{N}_1 \mid decide?(i) = yes \bullet ppant(i)\} \geq perc * nop \Rightarrow$$
$$\quad veto' = veto \setminus ppants?$$

$$\{i : \mathbb{N}_1 \mid decide?(i) = no \bullet ppant(i)\} \cap (veto \setminus ppants?) \neq \varnothing \lor$$
$$100 * \#\{i : \mathbb{N}_1 \mid decide?(i) = yes \bullet ppant(i)\} < perc * nop \Rightarrow$$
$$\quad veto' = veto$$

$$perc' = perc$$

3.4 Commissions

A group of participants *commission* is allowed to separate from the teleconferencing session, and to form a commission, which produces a separate product, independent from the product of the main session, Commissions are provided with their own arrow; in a sense, each commission is a teleconferencing session on its own. At any time, the participants of such a commission can decide to merge into the main session again (this will be described later on, in the schemas *LeaveCommission* and *EndCommission*). Each

commission is labelled by a natural number. By default, the label of the main session is 0.

The result of a commission *commission* (i) is stored in *session* (i), being a collection of data-files, while *noc* denotes the *n*umber *o*f *c*ommissions.

Commission

$$
\begin{array}{l}
Participant \\
commission : \mathbb{N} \rightarrowtail\!\!\!\!\!\!\twoheadrightarrow \mathbb{P}\, PPANT \\
session : \mathbb{N} \nrightarrow \mathbb{P}\, DFILE \\
noc : \mathbb{N} \\
\hline
0 \in \mathrm{dom}\, commission \\
noc = \#(\mathrm{dom}\, commission) - 1 \\
\mathrm{dom}\, commission = 0 \mathinner{\ldotp\ldotp} noc \\
\forall\, i : 0 \mathinner{\ldotp\ldotp} noc \bullet commission(i) \subseteq known \\
\forall\, i : 1 \mathinner{\ldotp\ldotp} nop \bullet \exists_1\, j : 0 \mathinner{\ldotp\ldotp} noc \bullet ppant(i) \in commission(j) \\
\forall\, i, j : 0 \mathinner{\ldotp\ldotp} noc \bullet i \neq j \Rightarrow commission(i) \cap commission(j) = \varnothing \\
\forall\, i : 1 \mathinner{\ldotp\ldotp} noc \bullet \#(commission(i)) \geq 1 \\
\mathrm{dom}\, session = 0 \mathinner{\ldotp\ldotp} noc
\end{array}
$$

3.5 The schema *Change*

Change

$$
\begin{array}{l}
Commission \\
change : \mathbb{N} \nrightarrow \mathbb{N} \\
arrow : \mathbb{N} \nrightarrow \mathbb{N} \\
tempsession : \mathbb{N} \nrightarrow \mathbb{P}\, DFILE \\
\hline
\mathrm{dom}\, change = 0 \mathinner{\ldotp\ldotp} noc \\
\mathrm{ran}\, change \subseteq 0 \mathinner{\ldotp\ldotp} nop \\
\mathrm{dom}\, arrow = 0 \mathinner{\ldotp\ldotp} noc \\
\mathrm{ran}\, arrow \subseteq 0 \mathinner{\ldotp\ldotp} nop \\
\mathrm{dom}\, tempsession = 0 \mathinner{\ldotp\ldotp} noc \\
\forall\, i : 0 \mathinner{\ldotp\ldotp} noc \mid change(i) = 0 \bullet tempsession(i) = session(i) \\
\forall\, i : 0 \mathinner{\ldotp\ldotp} noc \mid change(i) > 0 \bullet ppant(change(i)) \in commission(i) \\
\forall\, i : 0 \mathinner{\ldotp\ldotp} noc \mid arrow(i) > 0 \bullet ppant(arrow\ (i)) \in commission(i)
\end{array}
$$

The schema *Change* specifies the basic functions *change* and *arrow*, which are used to denote whether a product is being changed, or whether an arrow is being used re-

spectively. Moreover, it specifies the auxiliary function *tempsession*, which stores the temporary product of a commission during a period of change.

During a period of change in a commission *commission* (i), the value of *change* (i) equals the label of the participant that is changing the product of this commission. If nobody is changing the result of *commission* (i), then *change* (i) has the value 0.

Even so, if a participant of *commission* (i) is using the arrow, then *arrow* (i) has the value of this participant. If nobody is using the arrow of *commission* (i), then *arrow* (i) has the value 0.

3.6 Beginning and end of a session

The schema *StartSession* describes the start of a teleconferencing session, when there are no participants yet, and the result is empty. Furthermore, nobody is busy changing the result of the session, nobody has the right of veto, the arrow is not claimed, and there are no commissions yet.

StartSession

Decision
Change

$nop = 0$

$ppant = \varnothing$

$dbase = \varnothing$

$veto = \varnothing$

$noc = 0$

$commission = \{0 \mapsto \varnothing\}$

$session = \{0 \mapsto \varnothing\}$

$tempsession = \{0 \mapsto \varnothing\}$

$change = \{0 \mapsto 0\}$

$arrow = \{0 \mapsto 0\}$

The schema *EndSession* describes the end of a session, when there are no participants left. The final result of the session is expressed by *endproduct*!. In order to ensure that a session cannot be ended before it has started, the extra condition $session \neq \varnothing$ has been added to the schema.

The expression $\Xi Change$ marks the occurrence of an operation which does not affect the state space of *Change*.

EndSession

$\Xi Change$

$endProduct! : \mathbb{P}\ DFILE$

$nop = 0$

$session(0) \neq \varnothing$

$endProduct! = session\ (0)$

3.7 Changing the product

The temporary result $tempsession\ (i)$ of $commission\ (i)$ can be changed by any participant in this commission. Recall that during this change, the value of $change\ (i)$ equals the label of this participant, while if nobody is changing the result of $session\ (i)$, then $change\ (i)$ has the value 0.

In the schema *BeginChange*, a member $ppant$ ($ppant?$) of $commission\ (comm?)$ starts changing the result of $session\ (comm?)$. At such a moment, no other participant is to be occupied with changing the result of the session, that is $change\ (comm?) = 0$. The function *change* assumes the label of the participant that is changing the product of the commission, so $change'\ (comm?) = ppant?$.

BeginChange

$\Delta Change$

$comm? : \mathbb{N}$

$ppant? : \mathbb{N}_1$

$ppant? \in \mathrm{dom}\, ppant$

$comm? \in \mathrm{dom}\, commission$

$ppant(ppant?) \in commission(comm?)$

$change(comm?) = 0$

$change' = change \oplus \{comm? \mapsto ppant?\}$

$arrow' = arrow$

$tempsession' = tempsession$

In the schema *EndChange*, a participant decides to stop changing the product of $commission\ (comm?)$. In this case, the participants in his commission together decide whether or not the changes produced during this last change are stored. If so, then the contents of $session\ (comm?)$ is replaced by the contents of $tempsession\ (comm?)$. If not, then the original product is preserved.

EndChange

$\Delta Commission$
$\Delta Change$
$\Xi Decision$
$comm? : \mathbb{N}$
$decide? : \mathrm{seq}_1\ YESNO$

$comm? \in \mathrm{dom}\ commission$

$\mathrm{dom}\ decide? = \{i : \mathbb{N}_1 \mid ppant(i) \in commission(comm?) \bullet i\}$

$change(comm?) > 0$

$change' = change \oplus \{comm? \mapsto 0\}$

$\{i : \mathbb{N}_1 \mid i \in \mathrm{dom}\ decide? \land decide?(i) = no \bullet ppant(i)\} \cap veto = \varnothing \land$
$100 * \#\{i : \mathbb{N}_1 \mid i \in \mathrm{dom}\ decide? \land decide?(i) = yes \bullet ppant(i)\} \geq$
$\quad perc * \#(commission(comm?)) \Rightarrow$
$\quad\quad session' = session \oplus \{comm? \mapsto tempsession(comm?)\} \land$
$\quad\quad tempsession' = tempsession$

$\{i : \mathbb{N}_1 \mid i \in \mathrm{dom}\ decide? \land decide?(i) = no \bullet ppant(i)\} \cap veto \neq \varnothing \lor$
$100 * \#\{i : \mathbb{N}_1 \mid i \in \mathrm{dom}\ decide? \land decide?(i) = yes \bullet ppant(i)\} <$
$\quad perc * \#(commission(comm?)) \Rightarrow$
$\quad\quad session' = session \land$
$\quad\quad tempsession' = tempsession \oplus \{comm? \mapsto session(comm?)\}$

$arrow' = arrow$

$commission' = commission$

$noc' = noc$

3.8 Using the arrow

A participant can start using the arrow, provided that no other participant is using it. The arrow $arrow\,(i)$ has the value j if participant $ppant\,(j)$ in $commission\,(i)$ is using the arrow, and it has the value 0 if nobody in $commission\,(i)$ is using the arrow.

In the schema *BeginArrow* it is decribed that a participant $ppant\,(ppant?)$ can start using the arrow of its commission $commission\,(comm?)$, provided that no other member of $commission\,(comm?)$ is using it, that is, provided that $arrow\,(comm?) = 0$. Since $arrow$ assumes the label of the participant that is using it, we have $arrow'\,(comm?) = ppant?$.

BeginArrow

$\Delta Change$
$comm? : \mathbb{N}$
$ppant? : \mathbb{N}_1$

$comm? \in \mathrm{dom}\, commission$

$ppant? \in \mathrm{dom}\, ppant$

$ppant(ppant?) \in commission(comm?)$

$arrow(comm?) = 0$

$arrow' = arrow \oplus \{comm? \mapsto ppant?\}$

$change' = change$

The schema *EndArrow* expresses that at any time a participant using the arrow can decide to stop using it.

EndArrow

$\Delta Change$
$comm? : \mathbb{N}$

$comm? \in \mathrm{dom}\, commission$

$arrow(comm?) > 0$

$arrow' = arrow \oplus \{comm? \mapsto 0\}$

$change' = change$

3.9 Removal and addition of a data-file

AddFile

$\Delta Change$
$comm? : \mathbb{N}$
$dfile? : DFILE$

$comm? \in \mathrm{dom}\, commission$

$change(comm?) > 0$

$tempsession' = tempsession \oplus \{comm? \mapsto (tempsession(comm?) \cup \{dfile?\})\}$

$change' = change$

$arrow' = arrow$

The schemas *RemoveFile* and *AddFile* describe how the temporary product in the session of *commission* ($comm?$) can be changed. In both cases, some participant is busy changing the contents of *tempsession* ($comm?$), so that *change* ($comm?$) > 0.

In the schema *AddFile*, a new data-file is added to the session, while in the schema *RemoveFile*, a data-file is removed from the session.

RemoveFile

$$\Delta Change$$
$$comm? : \mathbb{N}$$
$$dfile? : DFILE$$

$$comm? \in \operatorname{dom} commission$$
$$change(comm?) > 0$$
$$dfile? \in tempsession(comm?)$$
$$tempsession' = tempsession \oplus \{comm? \mapsto (tempsession(comm?) \setminus \{dfile?\})\}$$
$$change' = change$$
$$arrow' = arrow$$

The adaptation of a session is in fact a combination of the two schemas *AddFile* and *RemoveFile*. Namely, adaptation of a data-file can be described by removing the original file and adding its adapted version.

3.10 Starting and finishing a commission

BeginCommission

$$\Delta Commission$$
$$\Delta Change$$
$$comm? : \mathbb{P}\ PPANT$$

$$\forall i : 1 \mathinner{..} noc \bullet commission(i) \cap comm? = \varnothing$$
$$\#comm? \geq 1$$
$$noc' = noc + 1$$
$$commission' = commission \oplus \{noc' \mapsto comm?\}$$
$$session' = session \oplus \{noc' \mapsto session(0)\}$$
$$tempsession' = tempsession \oplus \{noc' \mapsto session(0)\}$$
$$change' = change \oplus \{noc' \mapsto 0\}$$
$$arrow' = arrow \oplus \{noc' \mapsto 0\}$$

At the start of a new commission, two directories *session* and *tempsession* are created, which are to contain the result and the temporary result (during a change of the product) of the commission. Moreover, the commission is provided with its own arrow, so the domain of *arrow* is extended. Even so, the domain of *change* is extended. Furthermore, in the schema *BeginCommission* it is described that participants can only

be member of one commission at a time, and each commission contains at least one member.

EndCommission

$\Delta Commission$
$\Delta Change$
$comm? : \mathbb{N}_1$

$comm? \in \mathrm{dom}\, commission$

$comm? > 0$

$\#(commission(comm?)) = 0$

$arrow(comm?) = 0$

$change(comm?) = 0$

$commission' = (\lambda\, i : 1 \mathinner{..} comm? - 1 \bullet commission(i)) \cup$
$\quad (\lambda\, i : comm? \mathinner{..} noc' \bullet commission(i+1))$

$session' = (\lambda\, i : 1 \mathinner{..} comm? - 1 \bullet session(i)) \cup$
$\quad (\lambda\, i : comm? \mathinner{..} noc' \bullet session(i+1))$

$tempsession' = (\lambda\, i : 1 \mathinner{..} comm? - 1 \bullet tempsession(i)) \cup$
$\quad (\lambda\, i : comm? \mathinner{..} noc' \bullet tempsession(i+1))$

$arrow' = (\lambda\, i : 1 \mathinner{..} comm? - 1 \bullet arrow(i)) \cup$
$\quad (\lambda\, i : comm? \mathinner{..} noc' \bullet arrow(i+1))$

$change' = (\lambda\, i : 1 \mathinner{..} comm? - 1 \bullet change(i)) \cup$
$\quad (\lambda\, i : comm? \mathinner{..} noc' \bullet change(i+1))$

The schema *EndCommission* describes how a commission is ended. A commission can only be ended if it contains no members. If a commission is ended, then the labels of the other commissions need to be relabelled, in order to ensure that the domain of *commission* equals $0 \ldots noc$.

3.11 Adapting a commission

The schema *ExtendCommission* describes that at any time, a participant in the main session can decide to join a commission.

ExtendCommission

$\Delta Commission$
$\Xi Change$
$comm? : \mathbb{N}_1$
$ppant? : \mathbb{N}_1$

$comm? \in \operatorname{dom} commission$

$ppant? \in \operatorname{dom} ppant$

$ppant(ppant?) \in commission(0)$

$arrow(0) \neq ppant?$

$commission' = commission \oplus$
$\quad \{0 \mapsto (commission(0) \setminus \{ppant(ppant?)\}),$
$\quad comm? \mapsto (commission(comm?) \cup \{ppant(ppant?)\})\}$

$session' = session$

$noc' = noc$

Conversely, the schema *LeaveCommission* describes that at any time, a participant in a commission can decide to join the main session again, provided that he is not busy changing the product, nor using the arrow of the commission.

LeaveCommission

$\Delta Commission$
$\Delta Change$
$comm? : \mathbb{N}_1$
$ppant? : \mathbb{N}_1$

$comm? \in \operatorname{dom} commission$

$ppant? \in \operatorname{dom} ppant$

$ppant(ppant?) \in commission(comm?)$

$change(comm?) \neq ppant?$

$arrow(comm?) \neq ppant?$

$commission' = commission \oplus$
$\quad \{0 \mapsto (commission(0) \setminus \{ppant(ppant?)\}),$
$\quad comm? \mapsto (commission(comm?) \cup \{ppant(ppant?)\})\}$

$session' = session$

$change' = change$

$arrow' = arrow$

$noc' = noc$

3.12 Invitation and withdrawal of a participant

During a session, a new participant can be invited to join the session. This feature is described in the schema *NewParticipant*. The (original) participants have to decide whether or not this new participant is to be allowed in the session. Furthermore, the new participant should agree to join the session, which is expressed by the input *agree?*. By default, the label of the new participant is $nop + 1$.

NewParticipant

$\Delta Participant$
$\Delta Commission$
$Decision$
$ppant? : PPANT$
$dbase? : \mathbb{P}\ DFILE$
$decide? : \text{seq}\ YESNO$
$answer? : YESNO$

$ppant? \notin known$

$\{i : \mathbb{N}_1 \mid decide?(i) = no \bullet ppant(i)\} \cap veto = \varnothing \land$
$100 * \#\{i : \mathbb{N}_1 \mid decide?(i) = yes \bullet ppant(i)\} \geq perc * nop \land$
$answer? = yes \Rightarrow$
$\quad nop' = nop + 1 \land$
$\quad known' = known \cup \{ppant?\} \land$
$\quad ppant' = ppant \oplus \{nop' \mapsto ppant?\} \land$
$\quad dbase' = dbase \oplus \{nop' \mapsto dbase?\} \land$
$\quad commission' = commission \oplus \{0 \mapsto (commission(0) \cup \{ppant?\})\}$

$\{i : \mathbb{N}_1 \mid decide?(i) = no \bullet ppant(i)\} \cap veto \neq \varnothing \land$
$100 * \#\{i : \mathbb{N}_1 \mid decide?(i) = yes \bullet ppant(i)\} < perc * nop \land$
$answer? = yes \Rightarrow$
$\quad nop' = nop \land$
$\quad known' = known \land$
$\quad ppant' = ppant \land$
$\quad dbase' = dbase \land$
$\quad commission' = commission$

$session' = session$

$noc' = noc$

In the schema *ExitParticipant*, it is described how a participant $ppant\,(i)$ can withdraw from the session at any time. The only condition is that $ppant\,(i)$ is not using the arrow, nor busy changing the result of the session. The labels of the other participants have to be changed, for labels greater that i, in order to ensure that the domain of $ppant$ is $\{0, ..., nop'\}$.

ExitParticipant

$\Delta Participant$
$\Delta Commission$
$Change$
$ppant? : \mathbb{N}_1$
$comm? : \mathbb{N}$

$nop > 0$

$ppant? \in \mathrm{dom}\, ppant$

$comm? \in \mathrm{dom}\, commission$

$ppant(ppant?) \in commission(comm?)$

$arrow(comm?) \neq ppant?$

$change(comm?) \neq ppant?$

$nop' = nop - 1$

$known' = known \setminus \{ppant(ppant?)\}$

$ppant' = (\lambda\, i : 1 \,..\, ppant? - 1 \bullet ppant(i)) \cup$
$\quad (\lambda\, i : ppant? \,..\, nop' \bullet ppant(i+1))$

$dbase' = (\lambda\, i : 1 \,..\, ppant? - 1 \bullet dbase(i)) \cup$
$\quad (\lambda\, i : ppant? \,..\, nop' \bullet dbase(i+1))$

$session' = session$

$noc' = noc$

4 Summary and Future Research

We have presented a specification in Z of the framework of a teleconferencing system. This specification has been based mainly on the experiences with two such systems, namely 'Hypermedia System' from Alcatel/SEL and 'Meeting' from the CWI. This work originates from the need of Alcatel/SEL to find a rigorous basis for the construction of its Hypermedia System.

Experiences with other teleconferencing systems are to provide information whether our specification is sufficiently general for practical purposes. There is of course a good chance that experience will learn that the specification is too limited or too vague at some points, where the specification needs to be refined. We trust however that such refinements will be extensions of the specification, which do not have a substantial impact on the specification itself.

Our specification may act as a platform to decide on a standard terminology for this field of research. Even more so, it may even serve as a first step towards a common framework for teleconferencing systems. Ideally, a new teleconferencing system can be built by plugging the desired special features into this framework.

References

[dGI90] **R. Di Giovanni and P.L. Iachini.**
HOOD and Z for the development of complex systems.
In D. Björner, C.A.R. Hoare, and H. Langmaack, editors, *Proceedings VDM 90: VDM and Z – Formal methods in software development, LNCS 428*, pages 262–289. Springer-Verlag, 1990.

[Gro92] **The HOOD Technical Group.**[2]
HOOD reference manual, issue 3.1.1, 1992

[HS90] **F. Halasz and M. Schwartz.**
The Dexter hypertext reference model.
In *Proceedings NIST Hypertext Standardization Workshop*, pages 95–133. Gaithersburg, 1990.

[JLM+93] **K.H. Jerke, A. Lesch, E.M. Melchior, H. Rößler, and P. Szabo.**
Hypermedia services for IBC.
In Alex Gallis, editor, *Proceedings of the RACE IS&N Conference*, Paris, 1993.

[LRS93] **A. Lesch, H. Rößler, and P. Szabo.**
Broadband hypermedia.
In *Proceedings ISSLS'93*, Vancouver, 1993.

[LS90] **A. Lesch and P. Szabo.**
Hypermedia approaches.
In *Proceedings of Tools for Knowledge Organization and the Human Interface*, Darmstadt, pages 183–189. Index Verlag, 1990.

[RT89] **T.W. Reps and T. Teitelbaum.**
The Synthesizer Generator: a system for constructing language-based editors.
Springer-Verlag, New York, 1989.

[Spi87] **J.M. Spivey.**
Printing Z with LaTeX.
Note, 1987.

[Spi88] **J.M. Spivey.**
Understanding Z: a specification language and its formal semantics.
Cambridge Tracts in Theoretical Computer Science 3.
Cambridge University Press, Cambridge, 1988.

[Spi89] **J.M. Spivey.**
The Z Notation: a reference manual.
Prentice Hall International, Hertfordshire, 1989.

[2] The HOOD Technical Group is an international committee which coordinates the evolution of the HOOD methodology. It comprises a person for each company interested in HOOD.

[Tem94] **N. Temme et al, editors.**
CWI Quarterly 7(1), March 1994.
Special multimedia issue.